2022 FORTEI-International Conference on Electrical Engineering

(FORTEI-ICEE)

PROCEEDING FORTEI-ICEE 2022

IEEE Conference Number #57243
IEEE Catalog Number CFP22Y38-ART
ISBN 979-8-3503-9798-7

Tanjungpinang, INDONESIA
11-13 Oktober 2022

Organized by:

FORTEI
Forum Pendidikan Tinggi Teknik Elektro Indonesia

UNIVERSITAS MARITIM RAJA ALI HAJI
BELAJAR DAN BERTANYA TIADA JEMU

Technical Co-Sponsor by:

IEEE
INDONESIA SECTION

IEEE PES INDONESIA
Power & Energy Society®

2022 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE) | 979-8-3503-9798-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/FORTEI-ICEE57243.2022.9972977

2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE)

IEEE Conference Number # 57243

Copyright and Reprint Permission.

IEEE Catalog Number Part Number CFP22Y38-ART

ISBN 979-8-3503-9798-7

2022 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE) | 979-8-3503-9798-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/FORTEI-ICEE57243.2022.9972920

2022 Fortei-ICEE Title List

2022 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE) | 979-8-3503-9798-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/FORTEI-ICEE57243.2022.9972948

2022 Fortei-ICEE Author Index

2022 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE) | 979-8-3503-9798-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/FORTEI-ICEE57243.2022.9972965

Putra, Bayu Arivia	Fortei-ICEE02_006-010
Putra, Bayu Devanda	Fortei-ICEE08_036-040
Rachmilda, Tri Desmana	Fortei-ICEE01_001-005
Riyadi, Munawar Agus	Fortei-ICEE10_047-051
Sakti, Setyawan P.	Fortei-ICEE20_097-101
Sakti, Widhi Winata	Fortei-ICEE15_072-075
Samat, Ahmad Asri Bin Abd	Fortei-ICEE04_016-021
Santoso, Didik Rahadi	Fortei-ICEE13_062-065
Saputra, Jeki	Fortei-ICEE13_062-065
Saputra, Muhammad Ifan	Fortei-ICEE07_032-035
Saputra, Tarissya Brilianna	Fortei-ICEE16_076-080
Sartika, Erwani Merry	Fortei-ICEE16_076-080
Setyawan, FX Arinto	Fortei-ICEE07_032-035
Shidiq, Mahfudz	Fortei-ICEE14_066-071
Sulistiyanti, Sri Ratna	Fortei-ICEE07_032-035
Syafaruddin	Fortei-ICEE09_041-046
Syahrial	Fortei-ICEE08_036-040
Utomo, Teguh	Fortei-ICEE14_066-071
Vresdian, Devan Junesco	Fortei-ICEE17_081-086 Fortei-ICEE18_087-090
Walidainy, Hubbul	Fortei-ICEE08_036-040
Wijono, Wijono	Fortei-ICEE12_057-061

Multiclass Classification of COVID-19, Pneumonia, or Normal Lungs Based on Chest X-Ray Images with Ensemble Deep Learning

Tarissya Brilianna Saputra
Electrical Engineering Department
Universitas Kristen Maranatha
Bandung, Indonesia
1822031@eng.maranatha.edu

Novie Theresia Br. Pasaribu
Electrical Engineering Department
Universitas Kristen Maranatha
Bandung, Indonesia
novie.theresia@eng.maranatha.edu

Audyati Gany
Electrical Engineering Department
Universitas Kristen Maranatha
Bandung, Indonesia
audyati.gany@eng.maranatha.edu

Jimmy Hasugian
Electrical Engineering Department
Universitas Kristen Maranatha
Bandung, Indonesia
jhasugian@maranatha.edu

Erwani Merry Sartika
Electrical Engineering Department
Universitas Kristen Maranatha
Bandung, Indonesia
erwani.ms@eng.maranatha.edu

2022 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE) | 979-8-3503-9798-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/FORTEI-ICEE57243.2022.9972884

Abstract—**Coronavirus Disease of 2019 (COVID-19) has a high transmission and death rate. It is important to diagnose COVID-19 accurately and distinguish it clearly from other common lung diseases, e.g., pneumonia. Both diseases are detectable from chest X-Ray images. Therefore, an ensemble deep learning model is applied for multiclass classification of COVID-19, pneumonia, or normal lungs based on chest X-Ray images. ResNet50, VGG16, and InceptionV3 pretrained CNN models are employed to form an ensemble model. The chest X-Ray images are preprocessed in three steps, i.e., cropping, resizing, and normalization. Then, the pretrained models are trained with a new classifier at the top layer of the model. After the classifier is trained, then the pretrained ResNet50, VGG16, and InceptionV3 are fine-tuned. Lastly, the decisions from each model are assembled using Soft Voting. The ensemble deep learning model which produces the best result, which is formed by combining pretrained and fine-tuned ResNet50, VGG16, and InceptionV3 models, results weighted accuracy of 0.9752, weighted sensitivity of 0.9612, and weighted specificity of 0.9804.**

Keywords— multiclass classification, COVID-19, pneumonia, ensemble deep learning, ResNet50, VGG16, InceptionV3

I. Introduction

Coronavirus Disease of 2019 (COVID-19) is an infectious disease caused by the severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2) virus that first appeared on November 17, 2019 in Wuhan, China. Based on data from the World Health Organization (WHO), in 2 years and 10 months, COVID-19 has caused more than 6 million deaths worldwide, and there are currently 611 million active cases worldwide[1]. COVID-19 has symptoms similar to other respiratory diseases, but it has faster transmission and can lead to death. Therefore, it is important to distinguish the infection of COVID-19 disease from other respiratory diseases, such as pneumonia.

Chest X-Ray imagery is commonly used by physicians to make diagnoses of lung infections, including pneumonia and COVID-19 [2], including a diagnosis of the severity of the infection[3]. Deep learning can be applied to multiclass classifications in distinguishing lungs infected with COVID-19, pneumonia, and normal lungs.

Several studies have been conducted to create a classification model for COVID-19, pneumonia, or normal based on chest X-Ray imagery. Panwar et al. employed three Convolutional Neural Network (CNN) models to classify COVID-19 disease, pneumonia, and others trained using 4,563 chest X-Ray images[4]. A four-layers CNN was utilized for the first model, AlexNet for the second model, and modification of model 1 with data augmentation for the third model. When tested with 300 test data, 99% accuracy was obtained for the "COVID-19" class, 98.33% for the "pneumonia" class, and 98.67% for the "healthy/other disease" class.

Hilmizen et al. proposed a multimodal deep learning model that combines various CNN pretrained models with transfer learning to diagnose COVID-19 pneumonia using 2,500 CT-Scan images and 2,500 chest X-Ray images [5]. DenseNet121, MobileNet, Xception, InceptionV3, ResNet50, and VGG16 algorithms are used, then combined several algorithms by means of feature concatenation. The model concatenating ResNet50 with VGG16 and the model concatenating DenseNet121 with MobileNet provide the best performance, both with 99.87% accuracy, 99.74% sensitivity, and 100% specificity. The single model that has the best performance is the VGG16 with an accuracy of 98.93%, followed by the ResNet50, DenseNet121, and InceptionV3, all of which results an accuracy of 98.27%. The performance of the results of this research has been very good, but in this research only a binary classification was carried out, namely infected or not infected with COVID-19. The dataset used can also be considered quite small.

Narin et al. compared various CNN pretrained models to perform binary classification between COVID-19 with normal, COVID-19 with bacterial pneumonia, and COVID-19 with viral pneumonia based on 7,406 chest X-Ray images [6]. The algorithms compared are ResNet50, ResNet101, ResNet152, InceptionV3, and Inception-ResNetV2. ResNet50 performed best with 96.1%, 99.5%, and 99.7% accuracy for normal COVID-19 classification, COVID-19 classification with bacterial pneumonia, and COVID-19 classification with viral pneumonia. InceptionV3

978-8-3503-9798-7/22/$31.00 ©2022 IEEE

delivers the second-best performance with 95.4%, 98.6%, and 97.7% accuracy. The resulting performance was already good, but in this research only binary classification was carried out and a single model was used.

Chaudhary and Qiang created a deep learning ensemble model consisting of three different pretrained models to classify COVID-19, pneumonia, or normal[7]. The dataset from the "PG Challenge – COVID-19 Detection via Chest X-rays" was used[8]. The pretrained models used are Efficient Net-B3 with noisy student technique, SE-Resnext, and Efficient Net-B3 with adversarial training technique. There was an accuracy of 95.92%, sensitivity of 95.92%, and specificity of 95.97% in the test set leaderboard in the challenge that Chaudhary and Qiang participated in. The dataset used is large enough and the performance is good enough, but it is smaller than previous studies.

In this research, an ensemble deep learning model was proposed to carry out a multiclass classification of COVID-19, pneumonia, or normal. A chest X-Ray images dataset from the "PG Challenge – COVID-19 Detection via Chest X-rays" consisted of 21,390 images with three classes, namely COVID-19, pneumonia, or normal. Three CNN models with three different pretrained base models were combined (ensembled) to make the final classification decision. The pretrained base models used in this study are ResNet50, VGG16, and InceptionV3. Decisions from each model are combined using Soft Voting to make the final classification decision.

II. Method

The system input is in the form of chest X-Ray imagery. Preprocessing is carried out for each image according to the needs of each model. After going through the preprocessing process, the CXR image is input to the ResNet50, VGG16, and InceptionV3 pretrained models. Then, the results are fed to the classifier to make a prediction. The predicted results of the three classifiers are combined by an ensembler to determine the final decision. The ensemble technique used is Soft Voting. The system block diagram is shown on Fig. 1.

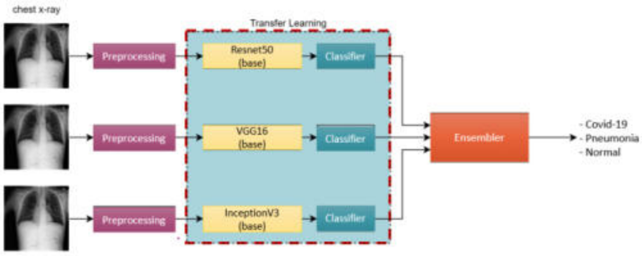


Fig. 1. Deep Learning Ensemble Model Block Diagram

The Chest X-Ray (CXR) dataset consists of 21,390 images with three classes: COVID-19, normal, and pneumonia. The dataset is divided into three parts, namely training data, validation data, and test data. The ratio used to divide the dataset is 80% training data, 10% validation data, and 10% test data. The largest portion is given for the training data so that the model can learn various variations in the image.

A. Preprocessing

There are three stages of preprocessing chest X-Ray image data carried out, namely cropping, resizing, and normalization. Before cropping, it is necessary to do image binarization. The binarization technique used is Adaptive Mean Thresholding. Then proceed to the Cropping process. To carry out the cropping process, scanning the binary image is performed to obtain the position of the white fringe border on the image. There are four positions obtained from scanning, namely the upper, lower, left, and right boundary positions. Implementation scanning to get all four limit positions. After obtaining all four boundary positions, the original image is cropped according to the position of the border of the periphery.

Image resizing is done at once by loading the image from the directory. Resizing is done by filling in the target_size arguments on the load_img function of tensorflow.keras.preprocessing.image according to the required size.

The normalization carried out is divided into two, namely zero-centering and scaling, depending on the needs of the model. Zero-centering is performed for resNet50 and VGG16 model inputs, while scaling is performed for InceptionV3 model inputs. Image normalization is implemented by using the preprocess_input function of each model from tensorflow.keras.applications. The following is an example of a preprocessing process for a pretrained model of ResNet50. See Fig. 2.

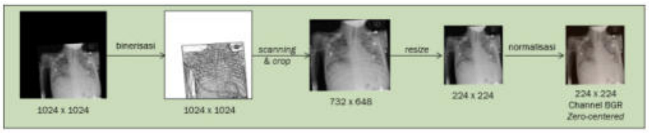


Fig. 2. Example of a Preprocessing Process For a ResNet50 Pretrained Model

B. Training Process

The model training process can be divided into several parts, namely model construction, model compile, and model fitting. In the model building section, the base model is defined, that is, the pretrained model. Then, a classifier is built on top of the base model. After the model is built, the trainable attribute of the base model is set to false so that the weight is not updated during training. Then, the model is compiled using adam optimizer with a default learning rate of 10-3, and sparse categorical cross-entropy loss. Then, the model is trained with training data and validation data using the fit function. Previously, some callbacks were defined to store training results and store models that have the highest validation accuracy.

For the fine-tuning stage, unfreeze several layers that you want to fine-tune by setting the trainable attributes for some of these layers to true. Then, the model is recompiled with Adam's optimizer with a learning rate of 10-4 and fitted back with training and validation data. For the resNet50 pretrained model, the unfreezeed part for fine-tuning is three blue conv5 blocks on the Fig. 3.

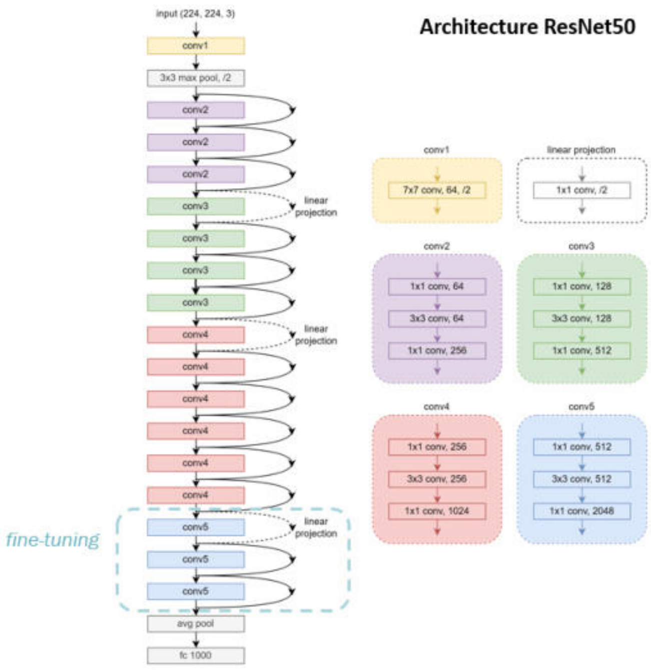


Fig. 3. ResNet50 Architecture Diagram (fine-tuning)

For pretrained VGG16 models, the unfreezeed parts are three purple 3x3 convolution blocks and one max pooling layer on Fig. 4.

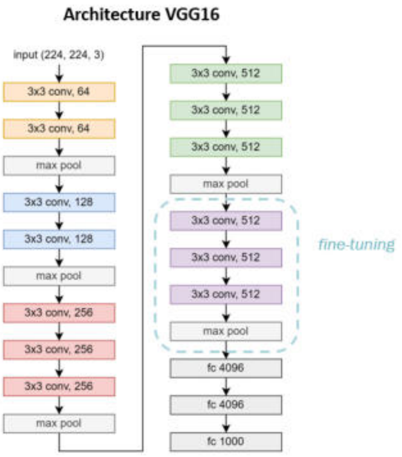


Fig. 4. VGG16 Architecture Diagram (fine-tuning)

For the pretrained model InceptionV3, the unfreezeed part is the mixed 9,10 block on the Fig. 5.

Several different classifiers and training schemes were used, resulting in 4 variations of models with pretrained ResNet50, 2 variations of models with pretrained VGG16, 2 variations of models with pretrained InceptionV3, and 2 variations of ensemble models. The combination of pretrained base type, classifier, and training scheme is shown in Table 1.

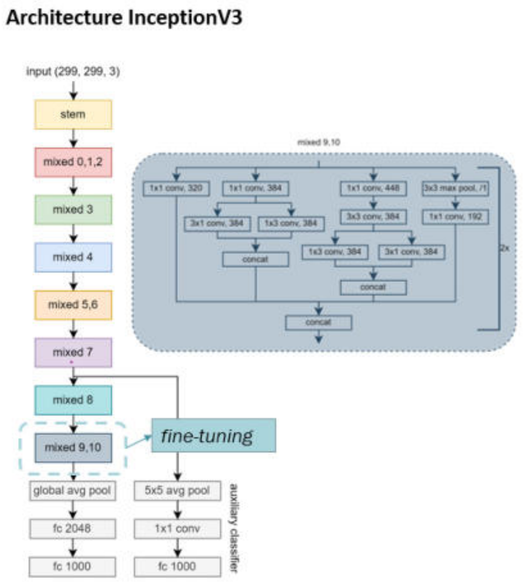


Fig. 5. InceptionV3 Architecture Diagram (fine-tuning)

TABLE I. COMBINATION OF PRETRAINED BASE, CLASSIFIER, AND TRAINING SCHEME OF EACH SINGLE MODEL TYPE

Model	*Pretrained base*			*Classifier*			*Training scheme*		
	ResNet50	*VGG16*	*InceptionV3*	*C1*	*C2*	*C3*	*S1*	*S2*	*S3*
Model-1	✓			✓			✓		
Model-2	✓				✓		✓		
Model-3	✓				✓			✓	
Model-4	✓				✓				✓
Model-5		✓				✓		✓	
Model-6		✓				✓			✓
Model-7			✓		✓			✓	
Model-8			✓		✓				✓

Here are the three types of classifiers used:

- Classifier C1: uses one global average pooling layer; one dense layer with 512 neurons, ReLU activation function, and Glorot initializer; one dense layer with 3 neurons and softmax activation function as output.
- Classifier C2: uses one global average pooling layer; one dense layer with 512 neurons, ReLU activation function, and initializer He; one Dropout layer; one dense layer with 3 neurons and softmax activation function.
- Classifier C3: uses one global average pooling layer; one dense layer with 256 neurons, ReLU activation function, and Initializer He; one Dropout layer; one dense layer with 3 neurons and softmax activation function.

Here are three types of training schemes used:

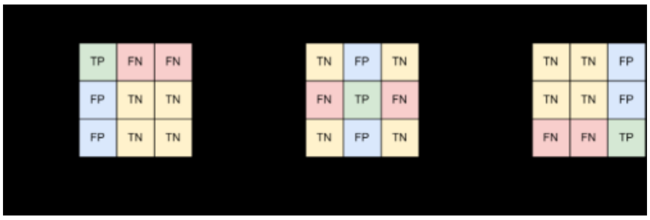
- Scheme S1: training is carried out only one stage, namely feature extraction only.
- Scheme S2: training is carried out in two stages, namely feature extraction and fine-tuning.
- Scheme S3: training is carried out in two stages like scheme S2, but the model with the best validation accuracy in the first stage of training is used to be fine-tuning in the second stage. Saved the weight model with the best validation accuracy in the first stage, then after the first stage of training is completed, the model is loaded with the best model weight and proceeds to the second stage.

Model-1 and Model-2 are used as baseline models, that is, models with the simplest structure and parameters. Therefore, both models are not used for the ensemble. Two ensemble models were created from Model-3 to Model-8 called Model-9 and Model-10. Model-9 ensemble models are made from Model-3, Model-5, and Model-7, that is, models with the S2 training scheme of each type of pretrained base. Model-10 ensemble models are made from Model-4, Model-6, and Model-9, which are models with the S3 training scheme of each type of pretrained base.

C. *Testing Process*

Model testing is performed by measuring metrics from the model's predicted results on the test data. Prediction results are obtained using the predict function of the model.

There are different types of metrics for classification, including accuracy, sensitivity, and specificity. Metrics are calculated based on the four elements in the confusion matrix, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP is a class that is actually positive and predicted as positive, TN is a class that is actually negative and predicted as negative, FP is a class that is actually negative and predicted as positive, and FN is a class that is actually positive and predicted as negative. In the multiclass classification, the positions of TP, TN, FP, and FN vary for each class. Fig. 6 indicates the confusion matrix for three different classes.



TP	FN	FN
FP	TN	TN
FP	TN	TN

TN	FP	TN
FN	TP	FN
TN	FP	TN

TN	TN	FP
TN	TN	FP
FN	FN	TP

Fig. 6. Confusion Matrix for Each Class

Sensitivity is a metric that demonstrates the model's ability to correctly identify all positive samples. Sensitivity and specificity are commonly used metrics in screening tests for a disease. Specificity is a metric that indicates the ability of a model to correctly identify all negative samples.

D. *Ensembler*

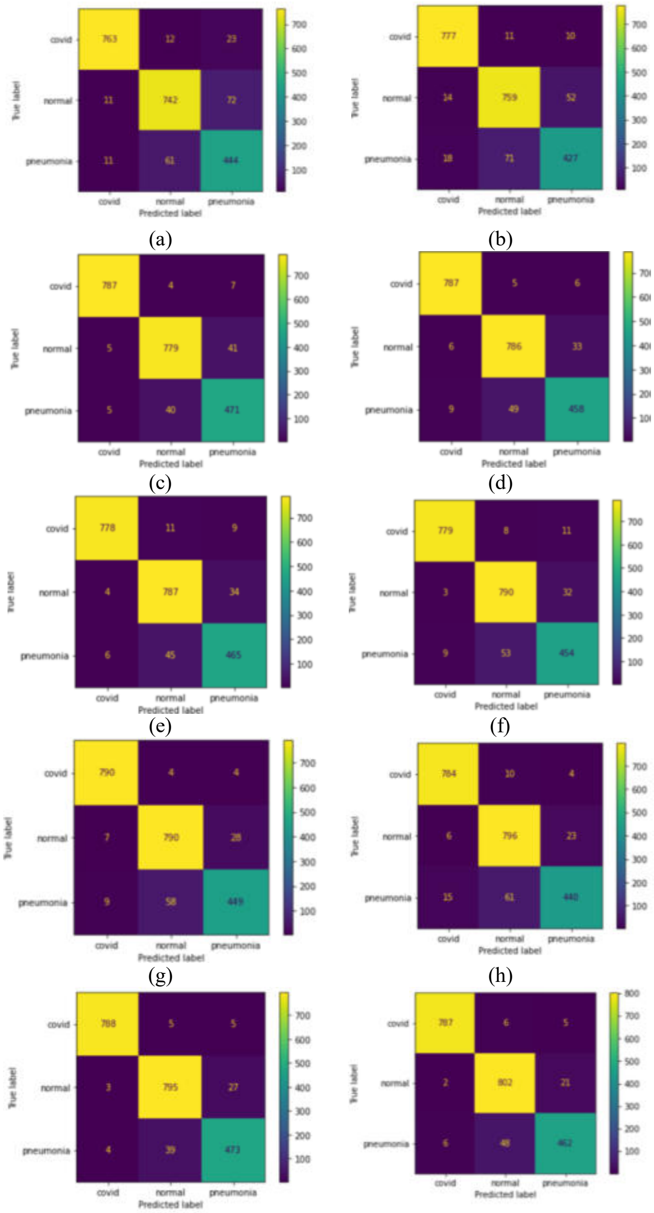
Ensemble learning is a learning technique that is carried out by training more than one model to complete the same job. Generally, the generalization ability of an ensemble model is stronger than its base models. One way to combine base learner model predictions in an ensemble is Soft Voting [9]. With the Soft Voting method, the output of each model, which is a probability, is averaged for the same class. Then, the average result of the $H^j(x)$ probability for each class is compared and the class with the highest probability average is selected as the final decision of the class. Thus, the output for the cj class can be formulated in equation (1).

$$H^j(x) = \frac{1}{T}\sum_{i=1}^{T} h_i^j(x) \qquad (1)$$

$h_i^j(x)$ is the output of the h_i classifier which is the probability for class c_j, $H^j(x)$ is the average probability for class c_j, and T is the number of classifiers.

III. Result and Discussion

The training process is carried out in 50 epochs for each stage of training. Models with the highest validation accuracy during training are stored and used for testing. Fig. 7 shows the confusion matrix of the Model-1 to Model-10 test results in the test data.

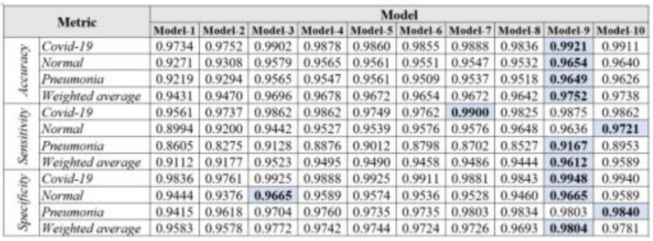


(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Fig. 7. Confusion Matrix Test Results of: (a) Model-1, (b) Model-2, (c) Model-3, (d) Model-4, (e) Model-5, (f) Model-6, (g) Model-7, (h) Model-8, (i) Model-9, (j) Model-10

A comparison of the metric results of the entire model in the test data can be seen in Table 2.

TABLE II. THE COMPARISON OF ALL MODEL METRICS ON TEST DATA



Metric		Model									
		Model-1	Model-2	Model-3	Model-4	Model-5	Model-6	Model-7	Model-8	Model-9	Model-10
Accuracy	*Covid-19*	0.9734	0.9752	0.9902	0.9878	0.9860	0.9855	0.9888	0.9836	**0.9921**	0.9911
	Normal	0.9271	0.9308	0.9579	0.9565	0.9561	0.9551	0.9547	0.9532	**0.9654**	0.9640
	Pneumonia	0.9219	0.9294	0.9565	0.9547	0.9561	0.9509	0.9537	0.9518	**0.9649**	0.9626
	Weighted average	0.9431	0.9470	0.9696	0.9678	0.9672	0.9654	0.9672	0.9642	**0.9752**	0.9738
Sensitivity	*Covid-19*	0.9561	0.9737	0.9862	0.9862	0.9749	0.9762	**0.9900**	0.9825	0.9875	0.9862
	Normal	0.8994	0.9200	0.9442	0.9527	0.9539	0.9576	0.9576	0.9648	0.9636	**0.9721**
	Pneumonia	0.8605	0.8275	0.9128	0.8876	0.9012	0.8798	0.8702	0.8527	**0.9167**	0.8953
	Weighted average	0.9112	0.9177	0.9523	0.9495	0.9490	0.9458	0.9486	0.9444	**0.9612**	0.9589
Specificity	*Covid-19*	0.9836	0.9761	0.9925	0.9888	0.9925	0.9911	0.9881	0.9843	**0.9948**	0.9940
	Normal	0.9444	0.9376	**0.9665**	0.9589	0.9574	0.9536	0.9528	0.9460	**0.9665**	0.9589
	Pneumonia	0.9415	0.9618	0.9704	0.9760	0.9735	0.9735	0.9803	0.9834	0.9803	**0.9840**
	Weighted average	0.9583	0.9578	0.9772	0.9742	0.9744	0.9724	0.9726	0.9693	**0.9804**	0.9781

Among all models, The Model-1 generates low metric values on the test data. This is because the model has experienced overfitting of the training data.

Based on the Model-1 and Model-2, it can be seen that the Model-2 using Dropout produces higher weighted accuracy and weighted sensitivity in the test data in Table 2. This shows that the use of Dropout and He Initializers decreases the overfitting experienced by the Model-1 and improves model performance in both validation and test data.

Model-4, Model-6, and Model-8, which are models that use the best model in the first stage of training for fine-tuning in the second stage of training, result in metric values that tend to be lower compared to the Model-3, Model-5, and Model-7 in the test data, as shown in Table 2.

Based on the four models that used the ResNet50 pretrained base, fine-tuning models (Model-3 and Model-4) resulted in better performance in the test data compared to models that were not fine-tuning (Model-1 and Model-2). Based on the performance of the Model-2 and Model-3 in the test data, the resulting metric increase in fine-tuning reached 0.0226 for weighted accuracy, 0.0346 for weighted sensitivity, and 0.0194 for weighted specificity.

The ensemble models, namely the Model-9 and Model-10, provide higher metric results than their base learner models in the test data, both in weighted accuracy, sensitivity, and specificity. The Model-9, compared to the Model-3, Model-5, and Model-7, resulted in an increase in weighted accuracy of 0.0056 to 0.008, weighted sensitivity of 0.0089 to 0.0126, and weighted specificity of 0.0032 to 0.0078. The Model-10, compared to the Model-4, Model-6, and Model-8, resulted in an increase in weighted accuracy of 0.006 to 0.0096, weighted sensitivity of 0.0094 to 0.0145, and weighted specificity of 0.0039 to 0.0088.

IV. CONCLUSION

A deep learning ensemble model has been successfully created by utilizing transfer learning of the pretrained ResNet50, VGG16, and InceptionV3 models to perform a multiclass classification of COVID-19, pneumonia, or normal lungs using chest X-Ray imagery. Model-9, which is a deep learning ensemble model, produced the most superior performance among all models with weighted accuracy of 0.9752, weighted sensitivity of 0.9612, and weighted specificity of 0.9804 in the test data. The Model-9 produced an increase in weighted accuracy of 0.0056 to 0.008, weighted sensitivity of 0.0089 to 0.0126, and weighted specificity of 0.0032 to 0.0078 from the performance of its base learner models, namely the Model-3, Model-5, and Model-7.

To develop model performance, it is recommended to increase the number of base learners in the ensemble to increase model diversity and optimize the p parameter on the Dropout to minimize overfitting.

REFERENCES

[1] World Health Organization, “WHO Coronavirus (COVID-19) Statistics.” https://covid19.who.int/ (accessed Jun. 19, 2022).

[2] S. Malik, S. Singh, N. M. Singh, and N. Panwar, “Diagnosis of COVID-19 Using Chest X-ray,” *Int. J. Informatics Inf. Syst. Comput. Eng.*, vol. 2, no. March 2020, pp. 55–64, 2021, [Online]. Available: https://ojs.unikom.ac.id/index.php/injiiscom/article/download/4024/2138.

[3] A. Rahman, S. M. Munir, I. Yovi, and A. Makmur, “The Relationship of Chest X-Ray in COVID-19 Patients and Disease Severity in Arifin Achmad General Hospital Riau,” *J. Respirasi*, vol. 7, no. 3, p. 114, 2021, doi: 10.20473/jr.v7-i.3.2021.114-121.

[4] A. Panwar, A. Dagar, V. Pal, and V. Kumar, “COVID 19, pneumonia and other disease classification using chest X-ray images,” *2021 2nd Int. Conf. Emerg. Technol. INCET 2021*, pp. 1–4, 2021, doi: 10.1109/INCET51464.2021.9456192.

[5] N. Hilmizen, A. Bustamam, and D. Sarwinda, “The Multimodal Deep Learning for Diagnosing COVID-19 Pneumonia from Chest CT-Scan and X-Ray Images,” *2020 3rd Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2020*, pp. 26–31, 2020, doi: 10.1109/ISRITI51436.2020.9315478.

[6] A. Narin, C. Kaya, and Z. Pamuk, “Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks,” Mar. 2020, doi: 10.1007/s10044-021-00984-y.

[7] S. Chaudhary and Y. Qiang, “Ensemble deep learning method for Covid-19 detection via chest X-rays,” pp. 1–3, 2022, doi: 10.1109/ee-rds53766.2021.9708581.

[8] M. A. Akhloufi and M. Chetoui, “Chest XR COVID-19 Detection,” 2021. https://cxr-covid19.grand-challenge.org/ (accessed Mar. 08, 2022).

[9] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, vol. 13, no. 2. CRC Press, 2012.